A POSTERIORI INDEXING, CLASSIFICATION AND RETRIEVAL OF TEXTUAL DATA

Noah S. Prywes, et al

Moore School of Electrical Engineering

Prepared for:

Office of Naval Research

January 1973

A POSTERIORI INDEXING, CLASSIFICATION AND RETRIEVAL OF TEXTUAL DATA

By January 1973

Noah S. Prywes, Allen L. Lang and Susan Zagorsky

Moore School of Electrical Engineering
University of Pennsylvania
Philadelphia, Pennsylvania 19104

N00014-67-A-0216-0014

INTRODUCTION

This paper reports on a series of programs that have been developed at the Moore School to process data-bases, consisting of textual items, and to index and arrange the data items in accordance with an automatically generated classification system. The programs produce the directories and the rearranged data-base on microfilm, where it may be searched using a microfilm reader, or magnetic tape for input to an on-line computer system for search and retrieval.

The only prerequisite to the use of the system is that the data-base be on computer-readable media, such as punched cards or magnetic tapes. The system was designed to be as self-contained as possible. Toward this end the indexing and classification relies exclusively on the content of the data-base itself, hence, the term a posteriori in the title, which denotes that decisions and processing occur only after the data base is made available. No prerequisities are imposed on the users, such as preparation of index term directories, establishment of categories or proofing and validation of data. The programs accept the data in fairly raw form, thereby differing from similar content-analysis and retrieval systems which require various inputs prior or external to the data-base being processed [1].

There are ample provisions in the course of the processing for the system to interact with the user, requesting corrections of data or directions regarding the use of index terms. In this latter sense, the system is semi-automatic. However, many provisions limit the guidance required from the user in the course of processing, so that he is not overwhelmed by a great amount of work.

By means of the directories that are generated and the ordering of the data in accordance with classification schedules the system is also highly efficient in subsequent retrieval. Once the directories have been generated, searches require only the processing of information in the directories, which eliminates the need for content analysis of the entire text in response to retrieval queries. Expanding or narrowing the searches for data can be fairly effectively carried out manually, either by using a microfilm reader with the directories and data-base placed on microfilm or by loading the data in the mass storage of an on-line computer and searching automatically.

The system aims to be useful to a wide variety of potential users. This includes researchers who review or survey literature to assemble textual data-bases of notes, abstracts or comments. Similarly, lawyers, authors and investigators in general compile information that they wish to arrange and retrieve later. Finally, the system is intended for libraries which maintain repositories of journals, articles, monographs, etc., which they wish to arrange on shelves or other storage media for subsequent retrieval. [2].

This paper describes the operation of the programs in performing two functions. The first is the automatic indexing of a data-base consisting of textual items. The second is the automatic generation of a classification system, including the assignment of classification numbers to text items, and, subsequently, the arrangement of the data-base in order of classification numbers of the items.

The input to the series of programs is, as indicated previously, a data-base consisting of text items. There are several products generated in the course of processing. The index terms are all extracted from the data-base, itself. The user, however, can control the process in many ways. One form of output consists of alphabetical lists of candidate index terms containing a variety of information on their frequencies of use, overall in the data-base and in specific sets of items. This type of information is used to find errors and to eliminate words, extracted from the texts, which are not suitable as index terms.

An output of the classification process is a two-part classification schedule. The index terms are herein referred to as <u>keys</u>, since they are used to indicate retrieval requirements. The canonical classification numbers are referred to as <u>nodes,</u> since they represent nodes in the hierarchical classification tree. The classification process divides the entire collection into progressively smaller groups of items which are considered by the system to be "alike," because they share index words.

The classification schedules then consist of a key-to-node directory and a node-to-key directory. Namely, one directory gives for each index term or key a list of classification numbers or nodes, each encompassing a group of "alike" items which share the respective index terms. Vice versa, the node-to-key directory gives the same information in an inverted manner, being ordered by classification numbers and giving all the corresponding keys for each node. These two directories are similar to classification schedules in other commonly used classification systems, such as Dewey or UDC where keys serve a function similar to subject headings.

The last product is the data-base, itself, in which the items are arranged in order of the classification numbers assigned to them. A data-base of some 1600 foreign broadcast items consisting of political and economic foreign news has been used to illustrate and test the operation of the system.

The narrowing or widening of a search for information involves lookup in the directories followed by a search in the ordered data-base, itself. As indicated, the directories and the data base can be produced by the computer on microfilm or on magnetic tape.

The programs have been developed by Allen L. Lang and Susan W. Zagorsky and are documented in a report entitled "Implementation of an Automatic, A Posteriori, Hierarchical Classification System" [3]. The programs have been prepared using FORTRAN. The computer used for the development is a Univac Spectra 70/46G with a Time Sharing Operating System. The source programs, being in FORTRAN, can readily be moved to another computer system. The programs and the documentation

could be made available to interested users by contacting the authors at the Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, Pennsylvania, 19104. The automatic classification methodology follows the work of Litofsky [4]. Research, expanding the capabilities of the system, continues at the Moore School.

INDEXING

Initially all the words in the text items are scanned and extracted as candidates for use as keys (index words) to index the respective items. The objective of subsequent processing and interactions between the user and the automated system is the reduction of the number of words considered as keys. The final outputs of the indexing process are the sets of keys selected to index the respective text items in the data-base. This serves as an input to the subsequent classification process.

The number of candidates for keys has to be greatly reduced for several reasons. First, as indicated by Litofsky [4], the effectiveness of the eventual retrieval from the automatically classified data-base depends on the number of keys used to index a text item, and on the number of unique classification numbers assigned to text items (in the subsequent classification process). Appropriate selection of the above enumerated parameters of the system can greatly increase the effectiveness of the automated retrievals. Normally, the initial number of candidate keys is far too large. Furthermore, for effective subsequent classification, keywords must denote the

"alike" nature of items that share these words as well as have the ability to discriminate between "unlike" items that do not share these words. Therefore, it is necessary to eliminate many key candidates which do not convey this information. Finally, assuming that the data-base being processed is in fairly raw form, it is necessary to correct errors and eliminate various message identifiers or other information which is not part of the text. Sometimes it is necessary to add or substitute indexing or identifying information.

The indexing process can be divided into two parts. The first part consists of standard techniques which are always applied in the beginning of the indexing process. The second part consists of user-system interactions which must be called for by the user.

In the first part, all text items are scanned. The user can indicate to the system the symbols that will be recognized as word delimiters. For instance, blanks are typically indicated as word delimiters. This will mean that words connected with a dash (-) or colon (:) will form key phrases. Other words phrases can be recognized automatically by, for instance, selecting as key phrases the words between quotation marks or a series of adjacent words in text starting with capital letters.[1] Next, there is a process of stem analysis which drops the suffixes of words, which considerably reduces the number of unique candidate keys. Finally, the user provides a "stop list" of words which are rejected from further consideration. Stop lists

[1] these capabilities have been added recently by Kemal Koymen

are available in literature; they number from 200 to 500 words and typically reduce the number of keys further considered by one third.

The second part of the processing involves much interaction with the user. In this process the number of key candidates is further reduced. To take the example of the Foreign Broadcast Information data-base, sixteen hundred items were included in the data-base. The total vocabulary of index term candidates prior to the interaction with the user was approximately 20,000 words. This was reduced to approximately 5,000 words during interaction with the user.

The user can delete, add or correct key words as well as combine index words into phrases. Obviously, it would be an enormous amount of work to indicate individually the words involved; therefore, various automatic processes are available to make such selection on a mass basis.

One process attempts to find words which are similar. The user can define the similarity in terms of position and number of characters in a word. Differences can be defined in a similar manner. The resulting lists are helpful in locating and correcting errors. The user can also require lists of words combining alphabetic and numeric characters.[1] These words frequently must be dropped.

Another important technique is to provide the user with listings of key words ordered by the frequency of the use in the data-base, or by the frequency of the documents using these words. The very highest and lowest frequency words must be scrutinized as candidates for elimination or

corrections. Finally, listings of keywords and frequencies in items having the largest numbers of candidate keywords must be given special attention. The user can delete or correct high and low frequency words in the respective items or overall in the data-base.

Thus, by submitting to the user lists of candidate keywords with the associated information on frequencies of usage, similarities, etc., the user can, with relatively little labor, reduce considerably the number of keys used to index the data-base. The eliminated words can be retained so that they can be used in processing subsequent data-bases.

Development work is currently underway at the Moore School to combine automatically index words into word phrases. A number of techniques are employed for this purpose. The user, himself, can identify these phrases. The phrases can also be identified automatically by considering frequency of occurrence in sentences and by use of special directories, such as in connection with times or dates. Finally, a study is underway which explores the effectiveness of processing relatively simple phrase structure analysis to identify noun phrases.

As indicated, the final outputs of the indexing process are sets of remaining keys and the item-identifications to which they have been assigned as index terms. This information is generated in a format directly acceptable by the automatic classification process described

below.

CLASSIFICATION

The automatic classification process involves a number of passes over the data; in each pass, the sets of keys and the respective text items are scanned and assigned to one of n groups. The first pass divides the data-base into n parts. In each additional pass, each previously obtained group is further sub-divided. The groups generated in every pass have each approximately the same number of unique keys used to index the respective text items. The unbalance, or differences in number of keys between groups, can also be specified by the user. This process can be repeated a specified number of times or it can continue until the number of items in a sub-group is below a certain specified limit. This terminal group is referred to herein as a cell. The user can specify the number n of sub-groups into which each group is divided in each pass. He can further specify the size of the cell.

The algorithm used to sub-divide groups follows algorithms originally suggested by Dr. David Lefkovitz and subsequently adopted by Litofsky [4]. Ideally, the process should find the optimal groups of keys that are within the specified sizes and that have a minimum of common keys. However, in devising algorithms for performing the division process, one quickly realizes that the amount of computer time required to perform the process is related to how near optimal results one gets. The Lefkovitz-Litofsky algorithms appear to get fairly good results with reasonably short computer processing times.

The multi-pass automatic classification process creates a tree; each pass of the process creates one level in the tree. Initially, all the items can be considered to be located in the root node of the tree. The first pass then creates n_1 nodes by dividing the data base into n_1 groups. The second pass may create $n_1 \times n_2$ nodes and so on. The tree thus created is illustrated in Figure 1. Each node is then associated with a sub-part of the data-base and a sub-part of the keys vocabulary, consisting of the keys used to index the respective items. Each node can then be assigned a classification number based on its position. For instance, the node 1.2.3 would indicate a node in the third level of the tree [3 digit positions] which emanates from the root node (1) and from the 2nd node in the 2nd level of the tree (counting from left to right). A terminal node forms a cell and is not further sub-divided. The classification number of the terminal node is assigned to all the items in the cell.

Finally, keys can also be associated with nodes. The process of associating keys with nodes starts with assigning to each terminal node or cell all of the unique keys used in indexing the respective items in the cell. A key which is associated with all the nodes which emanate from a lower level node would then be transferred to the lower level node, and so on. For instance, a key would be associated with a root node of the tree only if it is used in items in all the terminal nodes of the tree.

The automatic classification process final outputs are

the two directories and the ordered data-base described in the introduction above. They are the key-to-node directory, the node-to-key directory and the data-base, itself, with the text items ordered in accordance with the classification numbers assigned to them. Each item is followed also by the list of index terms used to index the item. These three outputs can be prepared either in a microfilm magazine which can be searched and viewed with the use of a microfilm viewer, or on a magnetic tape ready for input into a time sharing system for automated search and display.

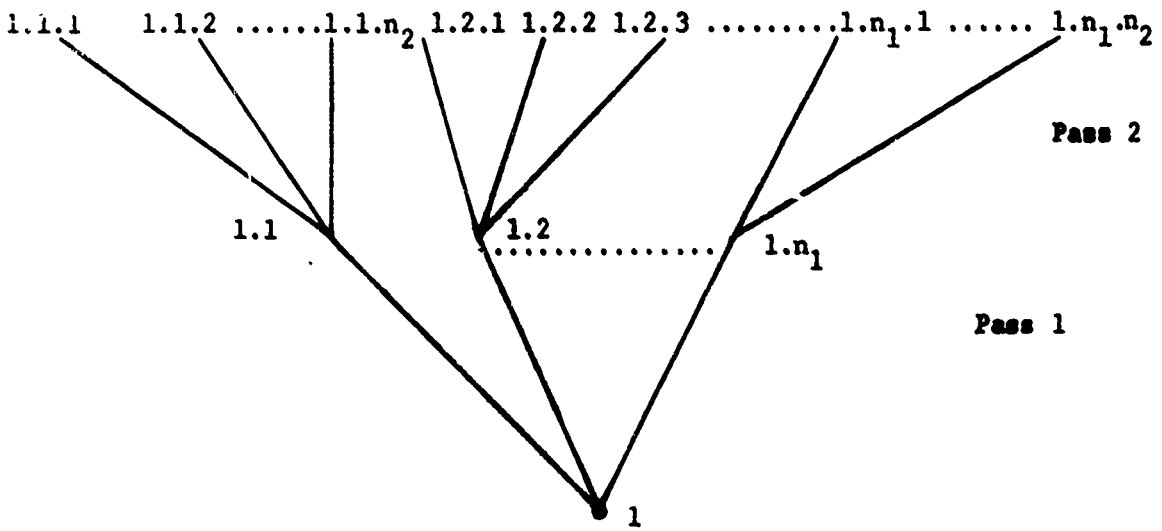

Figure 1

Illustration of Three Levels of a Classification Tree

SEARCH AND RETRIEVAL

A user desiring to retrieve text items from the data-base needs to make interactively repeated references to the directories and the ordered data-base. There are a number of possible starting points. A user can search the alphabetically ordered key-to-node directory for words to include in his retrieval specification. Or, he may search the node-to-key directory, proceeding systematically from the root node and selecting branches in the tree based on the keywords assigned to the nodes at the ends of such branches. The next step would be to look in the classification ordered data-base for the text items which have the classification numbers found in the directories.

The retrieval specification may be in the form of a logical expression involving conjunction, disjunction or threshold functions of keywords. These functions are easy to evaluate to find the respective classification numbers through a search of the key-to-node directory. For each keyword in the directory the corresponding classification numbers (nodes) are in ascending order (left adjusted), namely in accordance with the position in the tree, starting from the bottom level and ending at the top, and for each level starting from the left side and ending at the right side of the tree (see figure 1). To find the nodes corresponding to the conjunction of two keys, it is necessary to scan simultaneously the respective lists of nodes for the two keys. Because of the ordering of the nodes, only one scan is required. Only the pairs of classification numbers

taken from the two key entries, respectively, which are in a path from a root of the tree to a terminal node, satisfy the conjunction condition. Among such nodes, only the highest level node (having the larger number of digits) is retained for a look-up in the data-base. To perform a threshold operation, classification numbers for the minimum required number of keys must be in a single path from the root of the tree to a terminal node. The disjunction operation requires a similar look-up in the key-to-node directory. All the classification numbers found in the respective directory key entries are retained for further search in the data-base; however, when nodes are in a single path from the root of the tree to a terminal node, then only the lowest level node is retained for the look-up in the data-base.

A user may narrow or broaden his search by modifying these operations or by using the following broadening method. Once an item has been found to satisfy a retrieval specification, the other items in the same cell may be examined since these would have been recognized by the classification process as being "alike." Further expansion of the search may involve examination of the adjacent cells in the classification tree. An alternative is for the user to examine the keys of the cell and select from them additional keys to employ in his search, again using the key-to-node directory.

As indicated previously, these processes are very simple to execute and they can be performed manually. The search may then be confined to a data-base and directories placed in a magazine on microfilm. An automated

microfilm reader is needed to perform the search. Or, these functions may be processed on a time sharing system in an interactive mode.

CONCLUSION

The set of programs in the system described in this paper attempts to provide an economic solution to retrievals from a textual data-base. Costs of preparatory work to the user and of the data processing are fairly low as compared with those involved in more extensive text processing methods. The alternative is to perform syntaxic and semantic analysis, which would require both preparation of extensive directories, that do not exist yet, and a much greater investment in computer costs. Thus, this system is considered to be a compromise between cost and thoroughness, while retaining high effectiveness in retrieval.

REFERENCES

(1) P.S. Stone, et. al., "The General Inquirer: A Computer Approach To Content Analysis," The MIT Press, 1966.

(2) N.S. Prywes and B. Litofsky, "All-Automatic Processing For A Large Library," Session S70BB - Information Management Systems - Foundation and Future. In: American Federation of Information Processing Societies, Spring Joint Computer Conference, Atlantic City, New Jersey, 1970 Proceedings, Vol. 36, AFIPS Press, Montvale, New Jersey, 1970, 323-331.

(3) Allen L. Lang and Susan Zagorsky, "Implementation Of An Automatic, A Posteriori, Hierarchical Classification System," M.Sc. Thesis, The Moore School of Electrical Engineering, University of Pennsylvania, December 1972.

(4) B. Litofsky, "Utility Of Automatic Classification Systems For Information Storage and Retrieval," Doctoral Dissertation, University of Pennsylvania, 1969.